

# ハイブリッド発想支援システム「HIPS」

渡部 勇 三末 和男 新田 清 杉山 公造  
(株)富士通研究所 情報社会科学研究所

## 1 はじめに

我々は、これまでに、発散的思考支援システム Key-word Associator (以下 KA と略記) [1]、図的発想支援システム D-ABDUCTOR (以下 DA と略記) [2] の研究開発を進めてきた。

KA は、アイデアを広げていくきっかけとなるような刺激を、計算機から得るためのツールである。KA は、電子ニュース記事などのテキスト情報から、連想DBを自動的に構築する。そして、ユーザが入力した単語やテキストに対し、連想DBの中から関連する単語や関連するテキストを探し出し提示してくれる。システムから得られる情報がすべて役に立つわけではないが、中には自分だけでは到底思いつかないようなものも含まれている。これらが刺激となり、アイデアを広げていくこと、すなわち発散的思考を行なう際の助けになるわけである。

DA は、図を描き替えながら思考を展開していく過程を、計算機で支援するための対話型ツールであり、包含関係と隣接関係を表現可能な「複合グラフ」[3]を取り扱うことができる。それにより、アイデアをまとめていく収束的思考のための発想法として有名な KJ 法 [4] の過程を、計算機上で行なうことができるようになる。簡単な操作により図の高度な編集を行なうことのできる高機能グラフィックス・インタフェースを備えていることに加え、複合グラフを見やすく美しい図として描くための描画アルゴリズムを実装した自動描画機能を提供していることを特徴とする。

KA は発散的思考、DA は収束的思考というように、それぞれ異なる発想形態・思考過程を支援するシステムとして、独立に研究が開始され、進められてきたが、GrIPS (Group Idea Processing System) [5] の試作を機に、両者を統合する試みがなされてきている。

GrIPS とは、KA と DA に、「会議室機能」、「共有画面機能」などのグループウェア的な機能を加えたグループ発想支援システムである。GrIPS では、KA と DA を直列につなげることによって統合が行なわれている。すなわち、KA が発散的思考、DA が収束的思考という基本的な枠組は変えずに、KA での作業結果を DA で利用することによって、発想作業を全般に渡って支援するシステムとなっている。

この GrIPS における KA と DA の統合を第一段階とするならば、本稿で述べる HIPS (Hybrid Idea Processing System) は、いわば第二段階にあたる。HIPS では、単にツールをつなげるのではなく、それぞれの要素技術を組み合わせる形での統合が試みられている。そして、従来からの KA あるいは DA という枠組から離れることによって、

単体ではできなかった様々な支援機能を提供することを目指している。

以下では、まず、第2章でシステムイメージについて説明したあと、第3章で要素技術について、第4章で KJ 法支援への応用について述べる。また、第5章では、HIPS の実現可能性を調べるために行なった予備的な実験の結果について報告する。

## 2 システムイメージ

第1章で述べたように、GrIPS、HIPS は、ともに KA と DA を統合してできたものである。しかしながら、KA と DA の統合という点に関しては、その方法が全く異なったものになっている。ここでは、まず、GrIPS と対比しながら、HIPS がどのような構成になっているのかを説明し、その後、具体的な HIPS のシステムイメージについて述べることにする。

### 2.1 GrIPS と HIPS

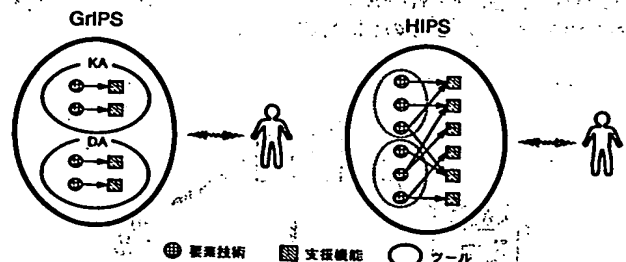


図1: GrIPS と HIPS

図1は、GrIPS と HIPS のシステムの構成方法を模式的に示したものである。

図に示すように、GrIPS は、複数のツールを組み合わせたものとなっている。なお、ここでは、簡単化のため KA と DA しか表していないが、この他にもグループ作業を支援する機能を持ったツールもいくつか組み込まれている。

GrIPS においては、KA は発散的思考を行なうフェーズ、DA は収束的思考を行なうフェーズで用い、基本的には両者を同時に使用することはない。KA と DA の間のインタラクションは、共有黒板機能を実現した SharedBoard と呼ばれるツールを介して行なわれる。すなわち、KA を用いながら SharedBoard 上で行なった発散フェーズの結果を、DA 上に COPY&PASTE によって移し収束フェーズ

を行なうカードを作成するという、単一方向の間接的な<sup>1</sup>データの流れが存在している。

GrIPS がツールレベルでの統合であるのに対し、HIPS は要素技術レベルでの統合になっている。したがって、KA、DA というツールレベルでの枠は存在せず、もともと KA の持っていた要素技術を収束的思考のための支援機能に利用したり、DA の持っていた要素技術を発散的思考のための支援機能に利用することもある。このように、両者の要素技術を組み合わせることにより、いままでにない高度な支援機能を提供することが可能になっていることが、GrIPS と HIPS のシステム構成上の大きな相違点になっている。

このように、KA、DA、GrIPS では、要素技術と支援機能の関係がほぼ一対一であったのに対し、HIPS ではこれが多対多の関係になっている。したがって、ユーザから実際に見える支援機能と、それを実現するためにある要素技術とを切り分けて考えることが重要である。

## 2.2 HIPS のシステムイメージ

我々は、まず、KA と DA の要素技術を洗いだし、それらを組み合わせた支援機能を考えるシーズ・オリエンテッドなアプローチと、発想作業の場面を想定しその中で求められる支援機能を洗いだし、それを実現するための要素技術を考えるニーズ・オリエンテッドなアプローチを並行して進めることにより、HIPS のコンセプト固めを行なった。

次に、このようにして得られた要素技術を整理・分類し、互いに直交する3つのグループに分けることにより、図2に示すような HIPS のシステムイメージが出来上がった。図2における各要素は、ツールではなく、要素技術の集合を表しており、図1の HIPS の中にある要素技術のまとめ(点線で囲まれているもの)に対応している。

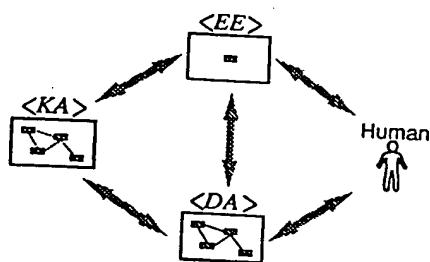


図2: HIPS のシステムイメージ

HIPS では、情報(アイデア)を、要素<sup>2</sup>とその間の関係<sup>3</sup>として捉える。

(EE) は、情報の要素を編集するための要素技術の集合であり、Element Editor という意味で (EE) と呼ぶことにする<sup>4</sup>。

<sup>1</sup>KA から DA に直接データを送ることも可能であるが、実際にはこのようにして使うことはあまりない。

<sup>2</sup>KJ 法におけるカードに相当する。

<sup>3</sup>カード間の関係に相当する。

<sup>4</sup>(EE)、(KA)、(DA) は仮称であり、まだ正式な名前は決まってい

(KA) は、情報の要素間の関係を抽出する要素技術の集合である。従来の KA が持っていた要素技術を中心に、新たなものをいくつか加えた形になっている。なお、ツールとしての KA と、ここでいう要素技術の集合としての (KA) を区別するために、括弧つきで表すことにする。

(DA) は、情報の要素間の関係を構造として捉え、その構造を編集・変換する要素技術の集合である。従来の DA が持っていた要素技術のうち、情報の要素の編集に関するものは、HIPS では (EE) の中に位置付けられる。それ以外の、構造のみに関するものに、新たな要素技術をいくつか加えたのが (DA) である<sup>5</sup>。

## 3 要素技術

本章では、第2章で説明した、(EE)、(KA)、(DA) という枠組にしたがって、HIPS における支援機能を実現するための、要素技術について順に述べていく。

なお、後の説明のため、各要素技術には記号を割り当ててある。また、KA、DA で実現されていた要素技術には「○」、新規に開発・適用されたものには「◎」、一部未実装のものには「△」、まだ実現されていないものには「×」のマークが、それぞれ付けられている。

### 3.1 情報の要素の編集: (EE)

情報を構成する要素を編集<sup>6</sup>するための要素技術群であり、扱う情報のタイプにより以下の3種類が用意される。

- E<sub>1</sub>: テキスト情報の編集
- × E<sub>2</sub>: イメージ情報の編集
- × E<sub>3</sub>: サウンド情報の編集

### 3.2 情報の要素間の関係抽出: (KA)

KA で実現されていた連想検索の技術を利用して、情報の要素間の関係抽出を行なう。連想DBの構築、連想検索、検索結果の変換の3つの要素技術群に分類される。

[連想DBの構築]

テキスト情報を入力とし、単語とテキスト<sup>7</sup>の間の関連度(連想の強さ)を統計的手法により計算し、連想DB(連想マトリックス)を自動的に構築する。

- Kb: 連想DBの構築

[連想検索]

連想DBを用いて、各種の連想検索を行ない、情報の要素の間の関係抽出を行なう。

4種類の検索パターン

- 1: テキストから関連テキストを求める
- 2: テキストから関連単語を求める
- 3: 単語から関連テキストを求める
- 4: 単語から関連単語を求める

と、3種類の検索タイプ(図3)

<sup>5</sup>ここでいう構造とは、いわゆる図解として表されるようなものだけを指しているわけではない。したがって、例えば、アウトライン・エディタのような技術要素もこの中に含まれる。

<sup>6</sup>情報の要素の入力・表示などのための要素技術も含まれる。

<sup>7</sup>情報の要素としてのテキスト。例えば KJ 法における一枚のカードがここでいう「テキスト」に相当する。

s: scalar を求める  
v: vector を求める  
m: matrix を求める

の組み合わせにより以下の 12 種類が用意される。  
なお、「→」の左側が入力、右側が出力を表す。

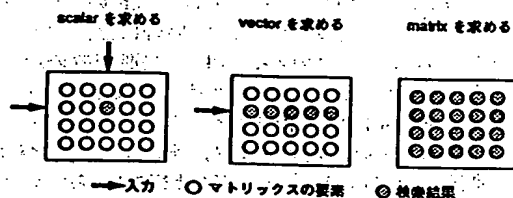


図 3: 検索タイプ

- △  $Ka_{1s}$ : テキストのペア → その間の関連度
- $Ka_{1v}$ : テキスト (群) → 関連するテキスト群
- $Ka_{1m}$ : → テキスト間の連想マトリックス
- △  $Ka_{2s}$ : テキストと単語のペア → その間の関連度
- $Ka_{2v}$ : テキスト (群) → 関連する単語群
- △  $Ka_{2m}$ : → テキストと単語の間の連想マトリックス
- △  $Ka_{3s}$ : 単語とテキストのペア → その間の関連度
- $Ka_{3v}$ : 単語 (群) → 関連するテキスト群
- $Ka_{3m}$ : → 単語とテキストの間の連想マトリックス
- △  $Ka_{4s}$ : 単語のペア → その間の関連度
- $Ka_{4v}$ : 単語 (群) → 関連する単語群
- △  $Ka_{4m}$ : → 単語間の連想マトリックス

#### [検索結果の変換]

連想検索で得られた結果の変換を行なう。

- $Kc_1$ : 検索結果の関連度による足切り・二値化など
- $Kc_2$ : テキスト間の連想マトリックスから、KJ 法のグループ化の作業順序に関する代用特性値を計算<sup>8</sup>

### 3.3 情報の構造の編集・変換: (DA)

DA で実現されていた図の操作・自動描画の技術を利用することにより、情報の構造の編集・変換を行なう。構造の編集、構造の変換、図のレイアウト、図のドレッシングの 4 つの要素技術群に分類される。

#### [構造の編集]

情報の要素であるノード<sup>9</sup>、情報間の関係を表すリンクの作成を行なう<sup>10</sup>。

- $De_1$ : ノードの作成
- $De_2$ : リンクの作成

<sup>8</sup> 第 4.7 節で説明する KJ 法グループ編成の順番ガイドのためのもの。連想マトリックスの要素 (関連度) の行方向の平均値、最大値などを利用する。

<sup>9</sup> ノードの中身の入力・編集に関する要素技術は (EB) の方にある。

<sup>10</sup> 構造の編集には、ユーザが行なうケースと、システムが自動的に行なうケースの両方がある。

#### [構造の変換]

情報の構造間の変換を行なう。ノード間の連想マトリックスから、ノードのグループを抽出するためのクラスタリングの技術などが、このカテゴリーに含まれる。

×  $Dc$ : クラスタリング

#### [図のレイアウト]

図のレイアウトを行なう。以下の 6 種類のレイアウト・パターンが用意される。

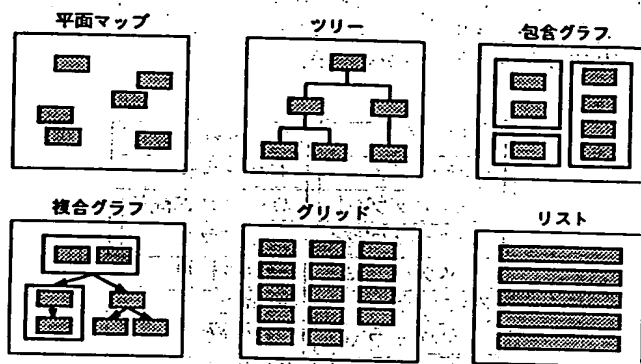


図 4: レイアウトのパターン

- $DI_1$ : 平面へのマップ<sup>11</sup>
- $DI_2$ : ツリーのレイアウト
- $DI_3$ : 包含グラフのレイアウト
- $DI_4$ : 複合グラフのレイアウト
- $DI_5$ : グリッドへのレイアウト
- $DI_6$ : リストとしてのレイアウト

#### [図のドレッシング]

ノードやリンクの可視性・サイズ・色などの視覚属性を変更することによって、図の「見かけ」を様々な形に変更する。

- △  $Dd_1$ : 拡大表示<sup>12</sup>
- ×  $Dd_2$ : ノードに関する量を色で表現<sup>13</sup>
- ×  $Dd_3$ : リンクに関する量を色で表現
- ×  $Dd_4$ : ノードのハイライトの連続表示<sup>14</sup>

<sup>11</sup> 林の数量化理論 IV 類 [6], Kruskal の方法 [7], spring アルゴリズム [8] などを用いる。

<sup>12</sup> DA では、複合グラフに関し、注目しているノードおよびその近辺を拡大しながら、同時に図の全体を表示する技術が実現されている。他のレイアウトアルゴリズムに対しても、それぞれの図の特性を生かした拡大表示のための技術が必要になる。

<sup>13</sup> 量の大きさにしたがって明るさ・濃さを変えたり、あるいは量の大きいものだけに色を付けるなど。

<sup>14</sup> 第 4.7 節で説明する作業の順番ガイドのためのもの。ノードのハイライトを時間とともに順に切り替えていく。

#### 4 KJ 法への応用

本章では、HIPS の提供する支援機能について述べる。ここでは、特に KJ 法の支援に焦点を絞り、KJ 法の各ステップに対し、どのような支援機能が提供されるのか、そして、その支援機能が、どのような要素技術の組み合わせで実現されているのかを順に説明していくことにする。また、HIPS という枠組の中で、従来からある KA、DA がどのような位置付けになるのかについても簡単に述べる。

##### 4.1 HIPS による KJ 法の支援

表 1 に、KJ 法のステップを示す。また、図 5 に、HIPS の提供する支援機能が、それぞれ KJ 法のどのステップを支援するものであるのかを示す。

ステップ 1	カード作成
ステップ 2	グループ編成
ステップ 2-1	カード初期配置
ステップ 2-2	グループ化
ステップ 2-3	ラベル付け
ステップ 3	図解化
ステップ 3-1	空間配置
ステップ 3-2	関係線付け
ステップ 4	文章化

表 1: KJ 法のステップ

図に示すように、HIPS の支援機能は、KJ 法 of 全ステップを網羅している。また、KA と DA を組み合わせた場合と比較すると、同一のステップに対し、レベルの異なる複数の支援機能が用意されているのが分かる。支援機能のレベルは、システムと人間が行なう作業の役割分担がに応じて、自動化を行なうレベル、ヒント・ガイドと提供するレベル、図の操作に関するインタフェースを提供するレベルの 3 種類に分類される。ユーザは、これらの支援機能の中から、問題の性質、カードの枚数、作業の行なう上での状況(時間的な制約など)、個人の嗜好などの諸々の条件に応じて、その場面にもっとも適したものを選択することが可能となる。

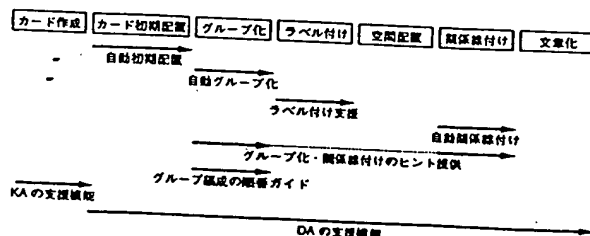


図 5: HIPS による KJ 法の支援

##### 4.2 自動初期配置

ステップ 2-1 の初期配置では、先入観に捕らわれずにグループ化を行なうため、ランダムに広げるのがよいといわ

れる。そして、広げられたカードを、まんべんなく繰り返しじっくりと読み、関係のある、似ているカードを集めていくステップ 2-2 を行なう。

カードの枚数がさほど多くない場合、あるいは、本格的な KJ 法を行なうことが必要な場合には、確かに上記のように進めるのが良いのであろう。では、大量のカードをあまり手間をかけずにまとめていきたい場合には、どうであろうか。ランダムに並んでいるよりは、初期配置の段階で関係のありそうなカードが近くに置かれている方が、グループ編成の作業を容易に短時間で行なうことができるだろう。特に、計算機上で行なう場合には、画面上に文字が読める大きさで同時に表示できるカードの枚数は限られたものとなるため、グループ化される可能性の高いカードが近くに置かれていることは、大きな支援となり得る。

自動初期配置は、カード間の関連度を  $Kb$ 、 $Ka_{1m}$  によって求め、 $DI_1$  を用いて関連が強いカードができるだけ近くに来るように配置する機能である(図 6(1))。

##### 4.3 自動グループ化

大量のカードに対応するためのもう一つのアプローチとして、ステップ 2-2 のグループ化までを自動的に行なってしまう支援が考えられる。自動グループ化は、自動初期配置と同様に、カード間の関連度を  $Kb$ 、 $Ka_{1m}$  によって求めたあと、 $Dc$  を用いてクラスタリングを行ない、関連が強いカードによるグループを自動的に作成する機能である(図 6(2))。

自動グループ化は、人間の手間を減らすという意味において、自動初期配置と比べ、より高度な支援を提供することになる。が、一方、大きな危険性を含んでいることも事実である。初期配置においては、そもそもランダムにすることが良いとされるぐらいであるから、システムの行なう配置にユーザの直観に合わない部分があったとしても、それが大きな問題になることはない。ところが、自動グループ化においては、システムによって直観に合わないグループ分けが行なわれた場合、それがそのまま最終結果に反映されることになってしまう。したがって、システムが行なったグループ化の結果をそのまま利用するのではなく、あくまでも初期状態として利用し、中身をチェックした上で、カードの移動、グループの分離、グループの統合などを行なうことが必要である。

##### 4.4 ラベル付け支援

ステップ 2-3 のラベル付けでは、ステップ 2-2 でできた各グループに対して、そのグループ全体を適切に表現するラベルを付けていく。このラベル付けの過程は、KJ 法の作業の中でもっとも難しい部分であるともいわれており、支援機能の実現が望まれるところである。

HIPS で提供されるラベル付けの支援は、グループ内にあるカードから重要な単語、すなわちキーワードを選びだし、ユーザに提示することによって行なわれる。キーワードの抽出は、 $Ka_{2v}$  に対して、グループ内のカード群を入力として与えることにより行なう<sup>15</sup>。抽出されたキーワードは、 $DI_6$

<sup>15</sup> もちろん、 $Ka_{2v}$  の前には  $Kb$  が必要である。

によって重要なものから順にリストの形で提示する、上位のものだけを選び、 $E_1$  に渡してユーザが文を作成する、あるいは、(ユーザの編集を経ずに) キーワードのセットとしてのラベルを自動的に付けるなどの方法によりユーザに提示される(図6(3)).

#### 4.5 自動関係線付け

ステップ2-2, 2-3を繰り返して行ない、全体が数個のグループになった段階で、ステップ3の図解化へと移行する。まず、グループの間の関係が適切に表現されるようにグループを配置し、続いてグループの構成要素(グループまたはカード)を同様に配置していく。このステップ3-1の空間配置が終わったら、次に、関係のあるカードやグループの間に線を引く、ステップ3-2の関係線付けを行なう<sup>16</sup>。

自動関係線付けは、カードおよびグループの間の関連度を  $Kb$ ,  $Ka_{1m}$  によって求め<sup>17</sup>、関連の強いペアの間に、 $De_2$  によって自動的に関係線で付ける機能である(図6(4))。ただし、関連が強い場合であっても、両者の間に包含関係がある場合には関係線は付けない。また、同一グループ内にあるカード・グループ間(兄弟の関係にあるもの)の場合にも、関係線を付けないようにするオプションも提供する必要がある。

システムによって付けられた関係線は、 $Dd_3$  を用い、関連度の大きさにしたがって線の色の濃さを変えるなどして、一目で関連の強さが分かるようなインタフェースによって提示される。また、適当な関連度の値で足切りし、上位のペアの間にのみ関係線を付けることも可能とする。この際、閾値をスライダなどで調節するインタフェースを提供することにより、関係線の数をユーザの望むレベルに合わせることができるようになる。

自動関係線付けの場合も、自動グループ化のときと同様に、システムが誤って付けた関係線が、そのまま最終結果に反映される可能性があるという意味での危険性があり、チェックを十分に行なうことが重要である。

#### 4.6 グループ化・関係線付けのヒント提供

自動グループ化・自動関係線付けは、カードおよびグループの間のグローバルな連想関係を用いて、グループ化・関係線付けのステップを自動的に行なうものである。基本的な部分はすべてシステムが行ない、ユーザは、クラスタリングのパラメータや関連度の閾値を適当に選んでグループの数や関係線の数を調節すること、そして、結果をチェック・修正することのみを行なう。

一方、グループ化・関係線付けのヒント提供の機能は、ある特定のカード・グループ(群)を中心としたローカルな連想関係を用いることにより、関連するカード群・グループ群を提示し、グループ化すべきもの、関係線を引くべきものをヒントとして与えるものである(図6(5))。

<sup>16</sup> グループの輪郭線は、システムによって自動的に描かれる。

<sup>17</sup> 各レベルのグループとカードをすべて  $Kb$  に送り関連度の計算を行なう。なお、グループに関するテキスト情報として、グループに付けられたラベルを利用する方法と、グループ内にあるすべてのカードをマージして利用する方法がある。

関連するカード群は、 $Kb$  に対して、すべてのカード・グループの内容をテキスト情報として渡して連想DBを構築するとともに、現在注目している(セレクトされている)カードあるいはグループが何であるかを指定して、 $Ka_{1o}$  の連想検索を行なう。

結果の提示方法としては、グループ化のヒントを提供する場合には、 $Dd_2$  により、関連度の大きさにしたがってカードの枠の色を変える、あるいは、関連の強い上位のものだけを選んでハイライトするといったインタフェースが提供される。関係線付けのヒントを行なう場合もほぼ同様で、注目しているカード・グループから関連するカード・グループの間に線を引きその色で関連度の大きさを表現する、あるいは、関連の強い上位のものとの間だけに線を引くといったインタフェースが提供される。

#### 4.7 グループ編成の順番ガイド

KJ法の完成図解を見てみると、第1回目のグループ編成によって、すべてのカードがグループ化されることはあまりなく、グループ編成の過程を何度か繰り返し、ある程度グループができてから初めてグループ化されるカードも数多く存在しているのが一般的である<sup>18</sup>。したがって、第1回目でグループ化されたもの、第2回目でグループ化されたものといった具合に、グループ化された順番にしたがって、カード間の順序付けを行なうことが可能である。ただし、同一サイクルでグループ化されたカード同士は同一の順位を持つものとする。

KJ法では、まず、すべてのカードを繰り返しじっくりと読む。そして、似ているカードのペアが見つかったらグループにまとめていく。カードの枚数が少ない場合には、この方法でもグループ編成を容易に行なうことができるが、枚数が増えてくるにしたがって、読むべきカードが増えるばかりでなく、繰り返し読む回数も増え、結果的に手間と時間が大幅に増大することになる。

ここで、仮に、上で述べたカードのグループ化に関する順序付けが、あらかじめ分かっているものとする。この情報をうまく利用すれば、初期の段階でグループ化すべきものから集中的に作業を進め、後の段階でグループ化されるものに関しては取り敢えず後回しにするといったような簡略化された方法を用いても、最終的なグループの構造の質をさほど落とすことはなく、グループ編成の過程を行なうことができるようになる。

グループ化に関する順序付けは、グループ編成の過程が終了した時点でないと分からない。したがって、グループ編成の順番ガイドの機能では、これに代わる代用特性を  $Kc_2$  により計算する。この代用特性を用い、 $Dl_5$  により、値の大きいもの(初期の段階でグループ化される可能性の大きなもの)から順にグリッド状にカードを配置する、あるいは、 $Dd_4$  により、値の大きいものから順にハイライトしていくといった形で、グループ編成の順番がガイドされる(図6(6))。

このガイド機能は、自動初期配置のような直接的な支援

<sup>18</sup> 必ずしもレベルを合わせながらカードを作成するわけではないため、異なるレベル・抽象度のカードが混在していることが原因であると思われる。

ではないが、その反面、先入観を与える可能性も少なく、また、本来の KJ 法の方法論から大きく外れるものでもない。

#### 4.8 KA と DA

KA および DA の要素技術は、一部拡張された形で、すべて HIPS の中に実現されている<sup>19</sup>。したがって、HIPS は KA および DA のスーパーセットになっており、従来 KA、DA が提供していた支援機能は、すべて HIPS の支援機能として提供される。

KA の支援機能は、HIPS においては、連想 DB を構築する  $Kb$ 、連想検索を行なう  $Ka_{1-4}$ 、結果を表示する  $DI_6$  によって実現される。 $DI_6$  (リストとして表示)の代わりに  $DI_1$  (平面マップとして表示)を用いて、検索入力を中心に、関連の強いものをその近くに配置するようにすれば、以前から課題として挙げていたリスト形式以外の方法による直観的な検索結果の提示が可能となる。

本章では、HIPS で新たに実現された支援機能として、KJ 法の各ステップにおける作業を自動化するもの、作業のヒント・ガイドを提供するものを説明してきたが、これらは、いわば付加的な機能であり、発想作業の主体が人間にあるという本質は変わっていない。したがって、DA で提供されていた図の操作に関する各種の機能は、HIPS においても重要な役割を演じる。特に、ステップ 3-1 (空間配置)とステップ 4 (文章化)に関しては、いまのところ HIPS で新たに用意された支援機能がないため、DA で提供されていた直接操作によるカードの移動機能、図解からのアウトライン作成機能などをそれぞれ用いることになる<sup>20</sup>。

#### 5 予備実験

HIPS は、KA と DA の要素技術をベースにしているが、既存の要素技術を別の目的に利用するケースや、新しい要素技術を開発・適用するケースも多々ある。したがって、そもそも、第 4 章で説明したような形で、シナリオ通りに支援機能を実現することができるかどうかについても不明な点も多い。支援機能が役に立つものであるかどうかを評価することも、もちろん重要であるが、その前に、その支援機能を実現するために用いられる要素技術が、それぞれ期待される結果を出せるかどうかを検証することが必要となる。

そこで我々は、要素技術に関し、様々な支援機能に用いられているもの、あるいは実験を行なわないと結果の予想が困難であるものから順に、実現性を検証するための予備実験を行なっている。本章では、現在までに得られている実験結果について簡単に報告する。

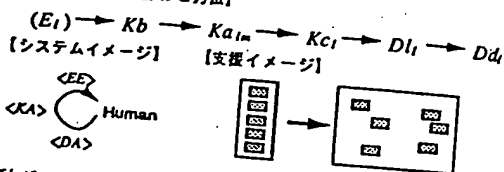
##### 5.1 カード間の関連度に関する実験

カード間の関連度を求めるため要素技術である  $Kb$ 、 $Ka_{1m}$  は、図 5 に示したように、多くの支援機能実現のために用いられている。

<sup>19</sup> 第 3 章において HIPS の要素技術の説明を行なったが、紙面の都合もあり、KA、DA の要素技術のすべてを挙げてはいない。  
<sup>20</sup> 空間配置に関しては、自動初期配置と同様の機能を提供することが考えられる。

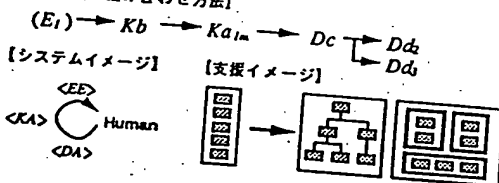
##### (1) 自動初期配置

【要素技術の組み合わせ方法】



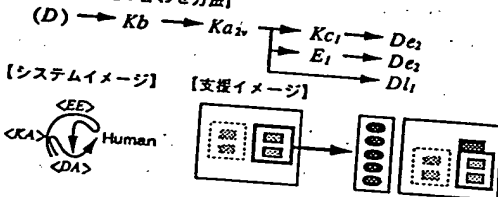
##### (2) 自動グループ化

【要素技術の組み合わせ方法】



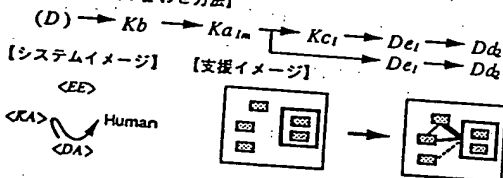
##### (3) ラベル付け支援

【要素技術の組み合わせ方法】



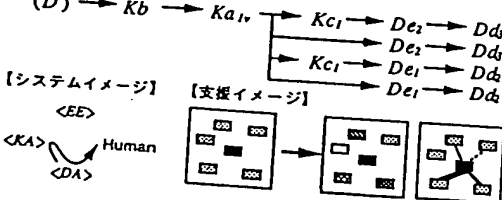
##### (4) 自動関係線付け

【要素技術の組み合わせ方法】



##### (5) グループ化・関係線付けのヒント提供

【要素技術の組み合わせ方法】



##### (6) グループ編成の順番ガイド

【要素技術の組み合わせ方法】

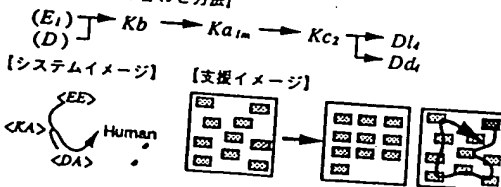


図 6: HIPS の支援機能

これらの技術に関しては、KA の応用実験の一環として、特許明細書の分類実験、電子マニュアルのリファレンス関係の抽出実験、事例ベースの検索実験を実施し、定量的な評価を行なっており<sup>21</sup>、実用レベルにあることが検証されている。ただし、KJ 法のカードを対象とする場合には、KA で想定しているものと比べて、テキストの数(カードの枚数)が少ないこと、一つ一つのテキストの長さ(カードに書かれている文章の長さ)が短いこともあり、同様の結果が得られるとは限らない。

そこで、すでに作成されている KJ 法の A 型図解の情報を用い、 $Kb$ 、 $Ka_{1m}$  によって求められるカード間の関連度と、図解におけるグループ化と比較する実験を行ない、カードのグループ化の自動化あるいはヒント提供の機能を実現するための関連度情報が得られるかどうかを調べた。

実験に利用した図解は、我々が作成したもの(テスト例 1)、KJ 法の解説書に出ていた例題でカードの枚数が少ないもの(テスト例 2)と多いもの(テスト例 3)の 3 種類である(表 2)。

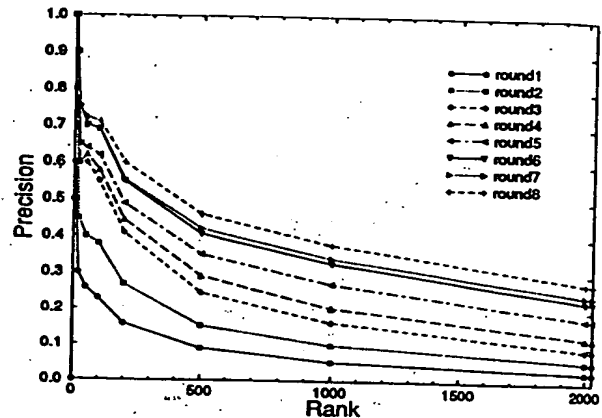
	カード枚数	平均テキスト長	グループ段数
テスト例 1	38 枚	21.4 文字	3 段
テスト例 2	25 枚	37.5 文字	4 段
テスト例 3	160 枚	38.2 文字	8 段

表 2: 実験に用いた図解

実験は以下の手順で行なった。まず、カード間の連想マトリックス(関連度のマトリックス)を求める。続いて、マトリックスの要素を関連度の大きさにソートし、カードのペアが、その間の関連度の大きさにしたがって並べられたリストを作成する。次に、各カードペアが、何回目のグループ編成によってグループ化されたものであるか、関係線によって結ばれているかどうかを調べていく。最後に、このリストのトップから第  $n$  位(Rank)までの範囲内での正答率(Precision)、すなわち実際にグループ化されていたペア、関係線で結ばれていたペアの割合を、 $n$  の値を 1, 2, 5, 10, 20, 50, 100, 200, ... というように次第に増やしながら計算していく。グラフ 1 は、テスト例 3 の情報を用いて得られた、グループ化との一貫に関する正答率をプロットしたものである。なお、グラフの 8 本の線は、それぞれ、第  $m$  回目( $m$  は 1 から 8)に作成されたグループの中にカードペアが存在すれば正解とする場合の正答率を表している。

グラフから明らかなように、リスト内での順位が高いほど、すなわち、関連度の値の大きいほど、そのペアが実際にグループ化されていた割合が高くなっている。テスト例 1 やテスト例 2 を用いた場合の結果も、これとはほぼ同じ形のグラフとなり、関連度の大きいものほど実際に行なわれたグループ化と一致するという傾向は、カードの枚数、カードに書かれているテキストの平均長さ、グループの段数、作成者のスキル(どの程度 KJ 法に精通し、忠実に行なっているか)などにはよらないことが分かる。

以上の結果から、次のように結論付けることができる。関連度の値の大きさペアが近くに配置されるようにカードの初期配置を決めれば、実際にグループ化される可能性の高いカードを近くに集めることができ、自動初期配置の機



グラフ 1: カード間の関連度とグループ化との一致度

能を実現することが可能となる。また、ユーザが指定したカードとの関連度の値が大きいカード群を提示することにより、実際にグループ化される可能性の高いカードをユーザに指示することができ、グループ化のヒント提供機能の実現が可能になる。

## 5.2 カードのマッピングに関する実験

カードを平面にマップする  $DI_1$  は、従来の KA, DA にはなかった要素技術であり、HIPS において新規に加えられたものである。現在は、マッピングアルゴリズムのベースとして何を用いたらよいのか、どのように改良していったらよいのかを検討している段階にあり、要素技術レベルでの評価実験を行なうことには大きな意味がある。

また、前節で述べたようにカード間の関連度の計算に関しては、ある程度のレベルの結果が得られることが分かってきた。したがって、 $DI_1$  の技術が完成すれば、自動初期配置の機能が実現され、支援機能全体としての評価実験も可能となる。

以上のような理由から、二次元マッピングに関する実験を行なっている。

実験には、KJ 法のデータではなく、電子マニュアル<sup>22</sup>から作成した連想マトリックスを利用している。これは、連想マトリックスと電子マニュアルのリファレンス関係の一致度を調べる実験を行なった結果、前節の KJ 法のデータを利用した場合と比較してより高い正答率が得られている点、電子マニュアルの方が、マッピングの結果をより客観的に判断することが可能である点を考慮した上で、要素技術のレベルの評価を行なうという点で、こちらの方が適当であると判断したからである。

図 7 に Kruskal の方法によるマッピングの例、図 8 に spring アルゴリズムによるマッピングの例を示す。

両者とも、多次元空間内での関係を二次元平面にマップすることによって生じる歪みが存在するため、連想マトリックスの関係をそのまますべて反映できるわけではないが、

<sup>21</sup> これらの実験結果に関しては、いずれ報告する予定である

<sup>22</sup> UNIX の /usr/man/man2 (システムコールに関するマニュアルページ) の情報を利用している

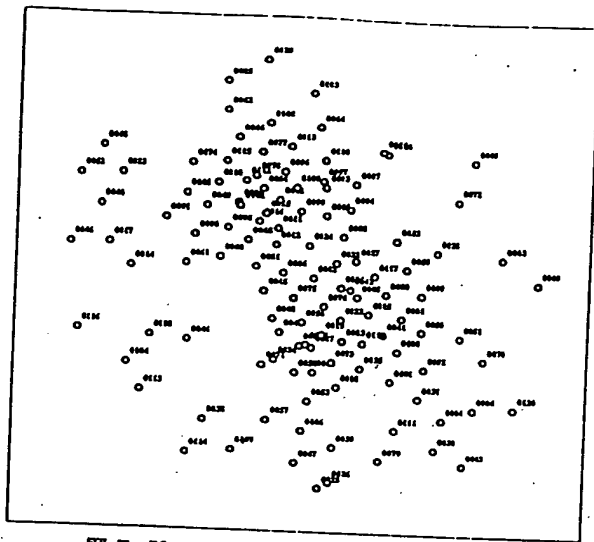


図 7: Kruskal の方法によるマッピング

基本的には近くに来るべきものが近くにくるような配置が得られている。

両者を比較すると、現在までに行なった実験の範囲内では、spring アルゴリズムの方がよい結果が得られている。また、図 7 は、どちらかというともんぺんなく散らばっているのに対し、図 8 では、ノードが集中したクラスターを見ることができる。内容をよく見てみると、グループとしてまとめられそうなものが実際にクラスターになっていることが分かる。

マッピングの結果に対する定量的な評価は、まだあまり十分に行っていない。したがって、上で述べたことは、あくまでも定性的な傾向であり、主観的な判断を含むものになっている。定量的な評価を行なうことは、今後の課題として重要なことである。そして、それと同時に、主観的な評価と定量的な評価をいかに結び付けていくかという点に関する考察を進めていくことが必要である。上でクラスターが見えると述べたが、もしクラスターに関する定量的な評価を行なうとするならば、まず、クラスターが見えるとはどういうことなのか、人間はいかにしてクラスターを認識しているのか、クラスターが見える図と見えない図の本質的な違いがどこにあるのかといったことを考えた上での評価が大切である。

## 6 おわりに

本稿では、KA と DA を統合した新しい発想支援システム HIPS について、そのシステムイメージ、要素技術、KJ 法への応用、予備実験の結果を説明した。

HIPS には、本稿で述べた以外に情報のブラウジングへの応用などが考えられている。また、要素技術、支援機能としてここで挙げた他にも様々なアイデアが出ている。本稿では、このうち、KJ 法の支援を行なうための機能と、それを実現するための要素技術を中心に報告した。

HIPS は、まだ発展中のシステムであり、未実装の要素

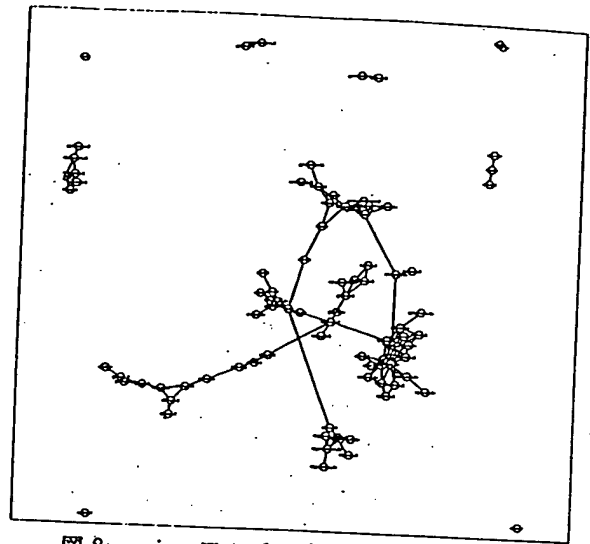


図 8: spring アルゴリズムによるマッピング

技術も多く、支援機能も完全に完成しているわけではない。したがって、今後は、まず、これらを実装・実現・洗練していく必要がある。また、第 5 章で述べたような要素技術レベルでの評価実験を進めていくとともに、支援機能のレベルでの評価、すなわち各機能が発想の支援として役に立つものであるかどうかについての評価も行なっていく予定である。

## 参考文献

- [1] 渡部 勇：発散的思考支援システム「Keyword Associator」第二版，計測自動制御学会第 15 回システム工学会部会研究会資料 (1994.7.18・東京)，pp. 9-16 (1994)。
- [2] 三末和男，杉山公造：図的発想支援システム D-ABDUCTOR の開発について，情報処理学会論文誌，Vol.35, No.9, pp. 1739-1749 (1994)。
- [3] 三末和男，杉山公造：図的思考支援を目的とした複合グラフの階層的描画法について，情報処理学会論文誌，Vol.30, No.10, pp. 1324-134 (1989)。
- [4] 川喜田二郎：KJ 法，中央公論 (1986)。
- [5] 神田陽治，渡部 勇，三末和男，平岩真一，増井誠生：グループ発想支援システム GrIPS，人工知能学会誌，Vol.8, No.5, pp. 65-74 (1993)。
- [6] 林知己夫，樋口伊佐夫，駒沢勉：情報処理と統計処理，産業図書 (1970)。
- [7] Kruskal, J. B.: Nonmetric Multidimensional Scaling: a Numerical Method, PSYCHOMETRIKA, Vol.29, No.2, pp. 115-129 (1964)。
- [8] Eades, P.: A Heuristics for Graph Drawing, Congressus Numerantium, Vol.42, pp. 146-160 (1984)。